III. Rejection under 35 U.S.C § 112

Claim 51 were rejected under 35 U.S.C. § 112, first paragraph, as containing subject matter which was not described in the specification in such a way to enable one skilled in the art to which it pertains to make and/or use the invention. This rejection is respectfully traversed.

The Applicant respectfully notes that amended claim 51 recites:

> The method according to claim 1, further comprising:
>
> transmitting a second data loader included in said bit stream at a transmission system;
>
> downloading the second data loader, at the receiver/decoder; and
>
> downloading the data loader and the data using the second data loader.

The Applicant respectfully points out that the phrase "at the receiver decoder" is a limitation of "*downloading* the second data loader," and has never been a limitation of "*transmitting* a second loader included in said bit stream." However, claim 51 has been amended to recite transmitting "at a transmission system" thereby clarifying the method to enable one of ordinary skill in the art to understand that transmitting may occur at a transmission system. Accordingly, withdrawal of this rejection is respectfully requested.

IV. Rejections under 35 U.S.C § 103

Claims 1, 2, 9, 20, 21, 26, 27, 31, 39, 43, 46, 51 were rejected under 35 U.S.C. §103(a) as being unpatentable over European Patent No. EP 0 680 213 A2 ("Menand," previously referred to as "Menand '213") in view of U.S. Patent No. 6,112,025

("Mulchandani"). Claims 1-3, 6, 20-22, 31, 39, 43, 44, 46, 49-53 have been amended in this reply to clarify the present invention recited. To the extent that this rejection may still apply to the amended claims, the rejection is respectfully traversed.

A. Menand & Mulchandani

The present invention relates to downloading a loader in native code, which in turn downloads software in native code. For example, claim 1 recites, "[a] method of downloading software in native code to a receiver/decoder, comprising the steps, at the receiver/decoder of: receiving a bitstream including the software in native code; downloading into the receiver/decoder a loader in native code for loading the software from the bitstream; downloading the software in native code into the receiver/decoder from the bitstream using said downloaded loader in native code, and storing said software in native code into the receiver/decoder."

Typically in a receiver/decoder, a bootstrap loader is used. However, the bootstrap loader cannot be modified once written into the read-only memory (ROM). Therefore, if the bootstrap loader becomes corrupted, it cannot be corrected. Similarly, if software to be downloaded by the bootstrap loader improves or changes structure, which is not recognizable to the bootstrap loader, the bootstrap loader cannot be updated.

Advantageously, the claimed invention provides a method, whereby a new loader is able to download improved software or software, which has changed structure. This software could be manufacturer firmware, the runtime engine, *etc.* Additionally, this software may be mainly in native code (but also could be in interpretative code). In the claimed invention, the bootstrap loader downloads an instream loader, *i.e.*, an additional loader from a bitstream, into random-access memory (RAM) of the receiver/decoder. The instream loader downloads the software from the bitstream and stores the software to hard disk. For example, software may be downloaded to update the firmware, the runtime engine, or resident software. The claimed invention allows downloading software, which has a specific structure.

In contrast, Menand states:

> [a] processor in an AVI [audio video interactive] receiver, under control of a system loader, first extracts the directory module from the data flow, and utilizes the information contained in the directory to determine which code module is first to be executed. That code module, called the autostart module, is then extracted from the data flow and loaded into the memory. When the autostart module has been completely loaded into memory, the processor begins to execute that code module. (Mendand col. 2, ll. 36-45)

In other words, the system loader as taught by Menand, first scans data components (or data flow) for a directory module, then extracts and loads the directory module. Using the loaded directory module, the system loader scans the data components for the first module, also termed an autostart module. Upon identification of the autostart module, this module is extracted and loaded. If the autostart module contains other code or chains to other modules, the system loader scans, extracts, and loads these associated modules.

For example, Menand states,

> [a]t initiation, a system loader executes API [application program interface] calls to the flow operating system...*These API calls from the system loader* initiate a scan of the data component packet service for the directory module...then the system loader initiates a scan of the AVI data component for the first module, termed an autostart module, which will initiate the autostart program. When the autostart module is located, it is extracted from the data component packet service and loaded into RAM 412...This program [the autostart module] may possibly load other code and data modules and chain to another module, *all via API calls* (Menand col. 7, ll. 28-51).

Therefore, the system loader, which loads the autostart module through API calls, also loads the other modules using API calls. Thus, Menand does not teach "downloading into the receiver/decoder a loader in native code for loading the software from the bitstream," as recited in claim 1, because a new loader is not being downloaded to download software.

That is, the autostart module as taught by Menand is not a new (or second) loader, in addition to the system loader, but simply is the initiating part of the application. Thus, the autostart module is (1) part of the application and (2) begins execution of the application; However, it is the responsibility of the system loader to scan, extract, and load subsequent modules. The system loader uses API calls or functions to load subsequent modules. For example, Menand specifically states, "[t] he function MODULE_CHAIN is used [by the system loader] to load a subsequent executable module, end execution of the current module, and begin execution of the newly loaded module at its entry point," (Menand col. 14, ll. 14-19). Therefore, the system loader as taught by Menand uses the API call, MODULE_CHAIN, to initiate a module, after the execution of the autostart module. Thus, the autostart module is *not* a new loader, which is required by claim 1, because the autostart module does not load subsequent modules.

Menand does not teach a downloading a new loader as required by claim 1. Additionally, Menand is completely silent as to downloading a loader in native code. Mulchandani fails to provide that which Menand lacks with respect to claim 1. Therefore, the combination of Menand and Mulchandani does not teach the claimed invention.

Mulchandani relates to a method for dynamic program linking. Menand characterizes the invention as:

> a system and method for dynamically linking a compiled procedure to referenced object components during execution of the compiled procedure. A native code loader loads the compiled procedure into a user's address space, and replaces a first instruction in the compiled procedure that references a first object component with a replacement instruction that invokes execution of a dynamic resolution procedure (Mulchandani col. 2, ll. 56-63)

The native loader method as disclosed by Mulchandani simply loads native programs into a user's address space. Mulchandani does not teach the downloading the loader as a second loader, as required by the claimed invention.

Because Menand and Mulchandani fail to teach downloading into a receiver/decoder a loader for loading software, and consequently, the other features of claim 1, claim 1 is patentable over Menand and Mulchandani. Independent claims 20, 39, and 52 have similar limitations and are also patentable over Menand and Mulchandani. Thus, claims 2, 9, 26, 27, 31, 43, 46, and 51, depending, directly or indirectly, are likewise patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

B. Menand & Mulchandani further in view of Bowen, Bestler, Metz, Jessup, and Hearing

As stated above, Menand and Mulchandani fail to teach the claimed invention as recited in amended claims 1, 20, 39, and 52. The remaining dependent claims were rejected under 35 U.S.C. §103(a) as being unpatentable over Menand and Mulchandani in view of U.S. Patent No. 5,367,571 ("Bowen"), U.S. Patent No. 5,608,732 ("Bestler"), U.S. Patent No. 5,666, 293 ("Metz"), European Patent No. 0 680 216 ("Jessup," previously referred to as "Menand '216"), or U.S. Patent No. 5,787,017 ("Hearing"). These references fail to provide that which Menand and Mulchandani lacks with respect to independent claims 1, 20, 39, and 52. These rejections are respectfully traversed.

With respect to Bowen, claims 3, 4, 22, and 23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Menand, Mulchandani, and Bowen. Bowen teaches a subscriber terminal with a plug-in expansion card, which is a non-volatile memory. Furthermore, Bowen is completely silent to a downloading a loader in native code, which in turn downloads software in native code, as required by independent clams 1, 20, 39, and 52. Therefore, Menand, Mulchandani, and Bowen fail to teach the claimed invention, and thus, clams 1, 20, 39, and 52 are patentable over Menand, Mulchandani, and Bowen, whether considered separately or in combination. Thus, claims 3, 4, 22, and 23, being dependent, are likewise patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

With respect to Bestler, claims 6 and 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Menand and Mulchandani, further in view of Bestler. Bestler relates to a mechanism for loading and replacing a record (or group of records) within a record set, or indicating that the record may be stored at the user's discretion, however, Bestler is completely silent to downloading a loader as recited in claims 1, 20, 39, and 51. Therefore, Menand, Mulchandani, and Bestler fail to teach the claimed invention, and thus, claims 1, 20, 39, and 52 are patentable over Menand, Mulchandani, and Bestler, whether considered separately or in combination. Thus, claims 6 and 25, being dependent, are likewise patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Regarding Metz, claims 29 and 49 were rejected under 35 U.S.C. §103(a) as being unpatentable over Menand and Mulchandani, further in view of Metz. Metz teaches "versioning" of operating systems, but is completely silent to downloading a loader as required by claims 1, 20, 39, and 52. Therefore, Menand, Mulchandani, and Metz fail to teach the claimed invention, and thus, claims 1, 20, 39, and 52 are patentable over Menand, Mulchandani, and Metz, whether considered separately or in combination. Thus, claims 29 and 49, being dependent, are likewise patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

With respect to Jessup, claims 28, 47, 48, and 50 were rejected under 35 U.S.C. §103(a) as being unpatentable over Menand, Mulchandani, further in view of Jessup. Jessup teaches a method for formulating an interactive TV signal. There is no disclosure or suggestion of downloading a loader as required by claims 1, 20, 39, and 52. Therefore, Menand, Mulchandani, and Jessup fail to teach the claimed invention, and thus, claims 1, 20, 39, and 52 are patentable over Menand, Mulchandani, and Jessup, whether considered separately in or in combination. Thus, claims 28, 47, 48, and 50, being dependent, are likewise patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Claim 13 was rejected under 35 U.S.C. §103(a) as being unpatentable over Menand, Mulchandani, further in view of Hearing. Hearing teaches a data acquisition apparatus and is completely silent to downloading a loader as required by claims 1, 20, 39, and 52. Therefore, Menand, Mulchandani, and Hearing fail to teach the claimed invention, and thus, claims 1, 20, 39, and 52 are patentable over Menand, Mulchandani, and Hearing, whether considered separately or in combination. Thus, claim 13, being dependent, is likewise patentable for at least the same reasons.

C. Jessup in view of Menand &Mulchandani

As stated above, the combination of Menand and Mulchandani, further in view of Jessup, fail to teach the present invention as recited in independent claims 1, 20, 39, and 52. Similarly, the combination of Jessup in view of Menand and Mulchandani also fail to teach the claimed invention. This rejection is respectfully traversed.

Claims 34, 35, 36, 44, 45, 52, and 53 were rejected under 35 U.S.C. §103 (a) as being unpatentable over Jessup in view of Menand and Mulchandani. As mentioned, Jessup teaches a method for formulating and interactive TV signal, but is completely silent to downloading a loader in native code, which in turn downloads software in native code. Again, Menand and Mulchandani fail to provide this feature. Therefore, Jessup Menand, and Mulchandani fail to teach the claimed invention, and thus, claims 1, 20, 39, and 52 are patentable over Jessup, Menand, and Mulchandani, whether considered separately or in combination. Thus, claims 34, 35, 36, 44, 45, and 53, being dependent, are likewise patentable for at least the same reason. Accordingly, withdrawal of this rejection is respectfully requested.

V. Concluding Remarks

Applicant believes this reply to be fully responsive to all outstanding issues and place this application in condition for allowance. If this belief is incorrect, or other issues arise, do not hesitate to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 11345.011001).

Respectfully submitted,

#45,079

Date: 4/24/03

Jonathan P. Osha, Reg. No. 33,986
ROSENTHAL & OSHA L.L.P.
1221 McKinney Street, Suite 2800
Houston, TX 77010

Telephone: (713) 228-8600
Facsimile: (713) 228-8778

46138_1